N-Tier DB Access

Lee Lueking

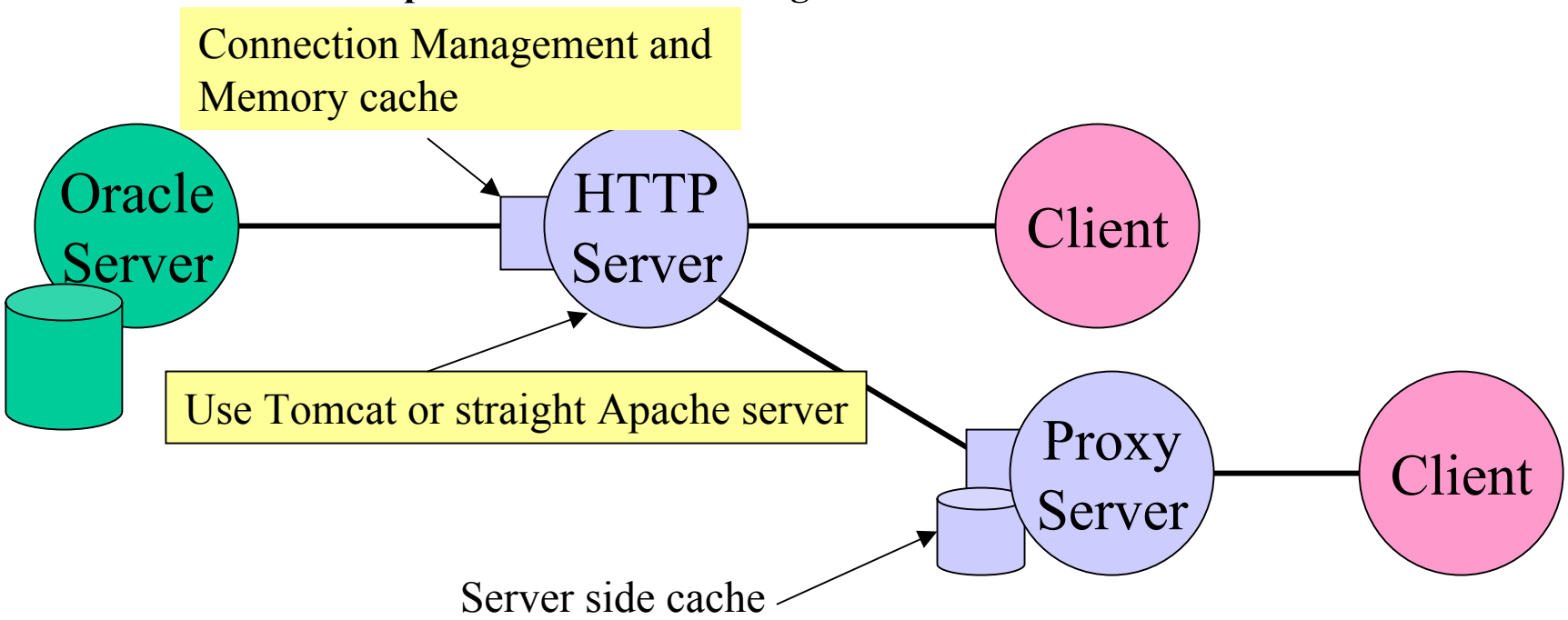
CMS Detector DB CD Briefing
Oct. 6, 2003

Motivation for Adding a Middle-tier (MT)

- **MT provides DB connection management**
- **Maintenance is simplified by decoupling DB and client interfaces. Releases, UPS products, etc. for client and MT are usually independent.**
- **Operational support is easier because one monitors a few MT servers, instead of hundreds of clients.**
- **DB access and load balancing polices can be added easily, as needed, to the MT.**
- **Excellent monitoring features available.**
- **Configuration flexibility.**
- **View in the DB can be different from view inside jobs without changing the client.**
- **On-demand population of the MT cache, and automated cache management, means "start it and forget it" operation.**
- **The caching drastically reduces bandwidth to oracle server and significantly lowers the latency for data access**
- **Network interruptions are less significant with the MT caching in place.**

Design Proposal Overview (CDF)

- **Build a middle-tier using:**
  - **Web services for MT-client communication with Client**
  - **Python or Java MT server using DAN (D0) experience for DB connection and caching**
  - **Create a simple client interface using HTTP.**



Getting started

- Starting with a simple prototype to test performance and scaling (see diagram on right)
  - Johns Hopkins is helping to do this work
  - Consulting from CEPA-APS.
- Understanding other projects which may be useful:
  - CMS has a similar framework called CLARENS used for data analysis which has ROOT and other modules. http://clarens.sf.net
  - Spitfire is a database access project in EDG WP2 that uses web services http://edg-wp2.web.cern.ch/edg-wp2/spitfire/